Lab Assignment
& Solution

Cybersecurity Professional Program

Introduction to Python
for Security

Network Communication

PY-06-LS3

Test Communication

Note: Solutions for the instructor are shown inside the green box.

Project Objective

Implement the code required to create client and server sockets, and to establish communication between them.

Lab Mission

Create a server socket and a client socket and enable them to communicate with each other.

Lab Duration

15 - 25 minutes

Requirements

- Basic knowledge of Python.

Resources

- Environment & Tools
    - Windows, macOS, Linux
        - PyCharm
        - Python 3

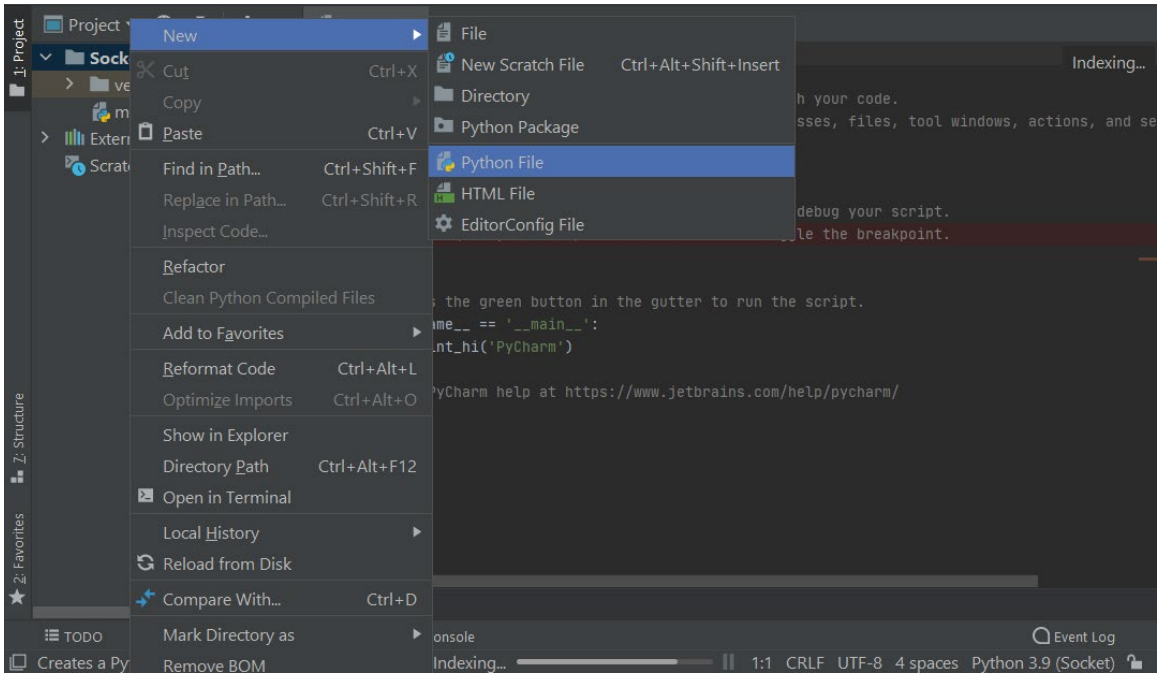
Textbook References

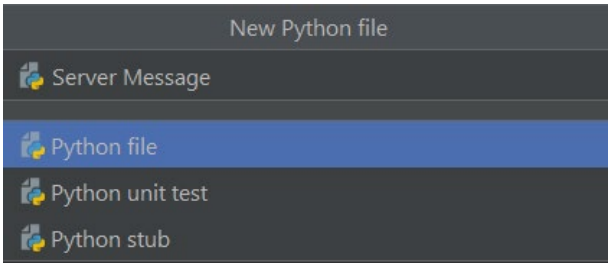
- Chapter 6: Network Communication
    - Section 3: Test Communication

Lab Task 1: Server Socket Creation

In this task, you will create a server socket to wait for a connection from a client, and send a message to it.

1 In the PyCharm project created in L1, create a new Python file by right-clicking the projects folder, and selecting **New → Python File**. Name the file **Server Message**.





2 Import the **socket** module to the file.

```
import socket
```

3 Create a socket variable.

```
import socket

sock = socket.socket()
```

4 Bind the socket to accept connections from all IP addresses on port 4444.

```
import socket

sock = socket.socket()

sock.bind(("0.0.0.0", 45000))
```

5 Allow only one connection to the socket.

```
import socket

sock = socket.socket()

sock.bind(("0.0.0.0", 45000))

sock.listen(1)
```

6 Allow the server to accept connections, and save the connection object and the address to variables.

```
import socket

sock = socket.socket()

sock.bind(("0.0.0.0", 45000))

sock.listen(1)

conn, addr = sock.accept()
```

7 Send a welcome message to the connected client.

```
import socket

sock = socket.socket()

sock.bind(("0.0.0.0", 45000))

sock.listen(1)

conn, addr = sock.accept()

conn.send("Welcome to the server!")
```

8 Encode the message sent to the client.

```
import socket

sock = socket.socket()

sock.bind(("0.0.0.0", 45000))

sock.listen(1)

conn, addr = sock.accept()

conn.send("Welcome to the server!".encode())
```

9 Accept a message from the client and print it.

```
import socket

sock = socket.socket()

sock.bind(("0.0.0.0", 45000))

sock.listen(1)

conn, addr = sock.accept()

conn.send("Welcome to the server!".encode())

print(conn.recv(2048).decode())
```

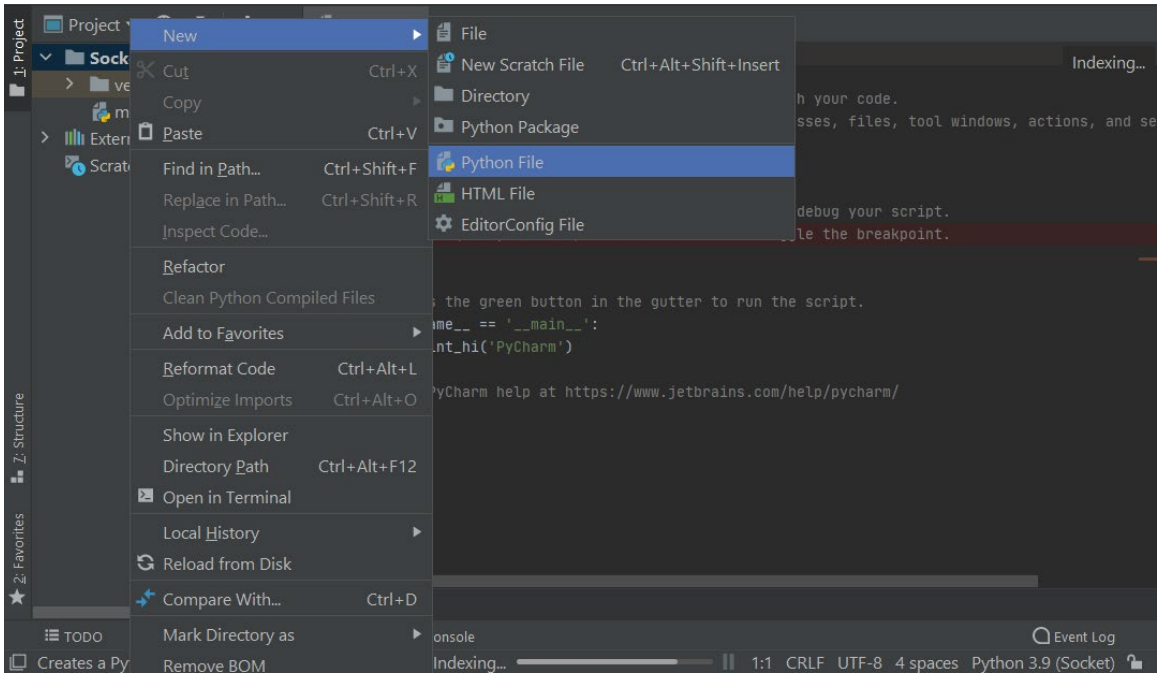
10 Close the connection.

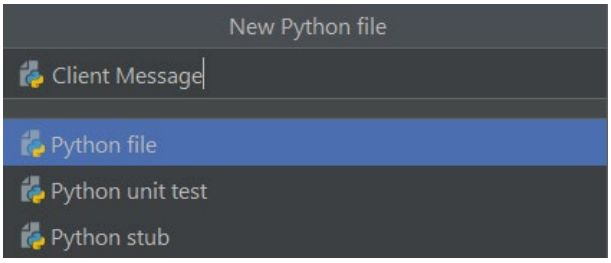
```
import socket

sock = socket.socket()

sock.bind(("0.0.0.0", 45000))

sock.listen(1)

conn, addr = sock.accept()

conn.send("Welcome to the server!".encode())

print(conn.recv(2048).decode())

sock.close()
```

Lab Task 2: Client Socket Creation

In this task, you will create a client socket to connect to the server, receive a message from it, and reply to the message.

1. Create a new Python file by right-clicking the projects folder, and selecting **New → Python File**. Name the file **Client Message**.





2. Import the **socket** module to the file.

```
import socket
```

3 Create a socket variable.

```
import socket

sock = socket.socket()
```

4 Connect the socket to the listener in the server.

```
import socket

sock = socket.socket()

sock.connect(("127.0.0.1", 45000))
```

5 Allow the client to receive the message from the server, and print it.

```
import socket

sock = socket.socket()

sock.connect(("127.0.0.1", 45000))

print(sock.recv(2048).decode())
```

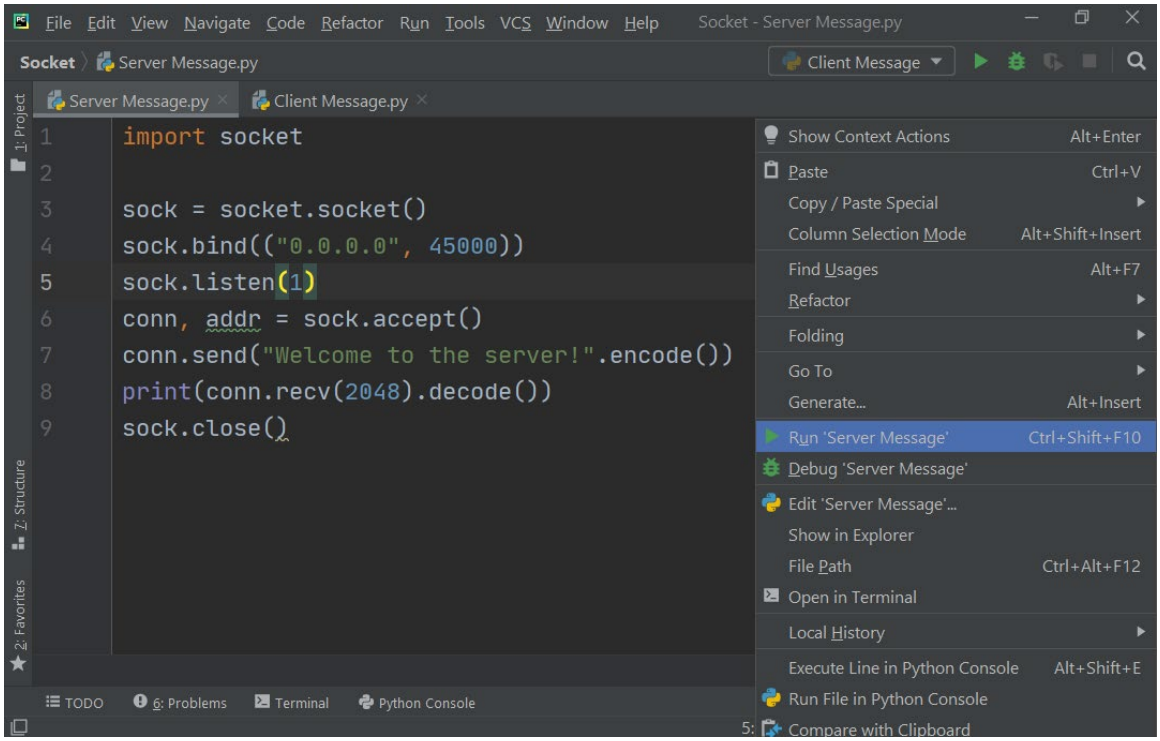
6 Return a message to the server notifying it that the client received the message.

```
import socket

sock = socket.socket()

sock.connect(("127.0.0.1", 45000))

print(sock.recv(2048).decode())

sock.send("Thanks!".encode())
```

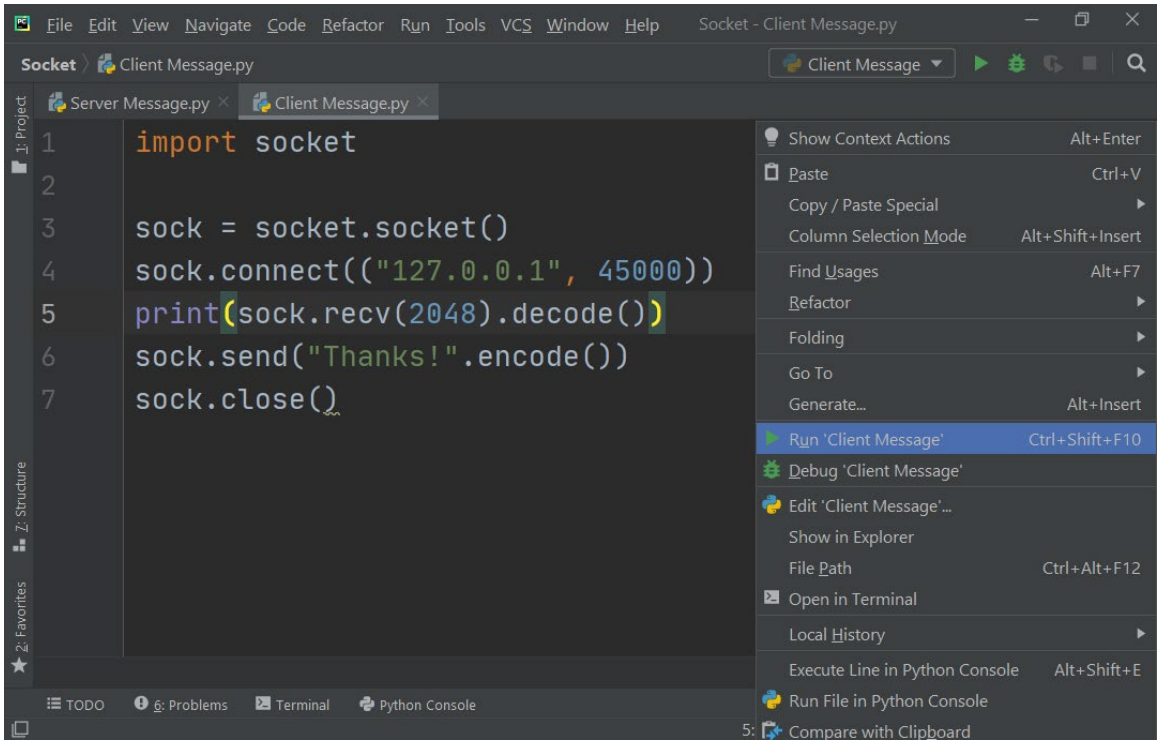
7 Close the connection.

```
import socket

sock = socket.socket()

sock.connect(("127.0.0.1", 45000))

print(sock.recv(2048).decode())

sock.send("Thanks!".encode())

sock.close()
```

8 Execute the server script.



9 Execute the client script.



10 Notice the message the client received.

11 Notice the message the server received.